Crypt of the Pixelmancer: Un videojuego de acción con IA modelando el comportamiento de los enemigos y generación procedural de mapas

Raül Salinas Natal

Resumen— La Inteligencia Artificial es una característica central en el mundo de los videojuegos ya que esta es la responsable, mediante el comportamiento de los enemigos, de generar la mejor experiencia de juego posible para el usuario. En este proyecto se ha buscado que el resultado final cuente con diversas funciones para generar una experiencia satisfactoria a la hora de jugar. La IA de los enemigos ha sido implementada de manera que sus patrones de comportamiento varíen en función del tipo de arma que lleve el jugador. Una pantalla de inicio con el título del juego e imágenes de los mapas y un selector de opciones interactivo dan la bienvenida a los jugadores al empezar una partida. Junto a ellas se ha implementado un apartado visual con sprites para el jugador con animaciones, sprites para enemigos, así como distintos fondos de nivel según el mapa seleccionado. En definitiva, el juego está listo para brindar muchísimas horas de diversión y un reto exigente a los jugadores más atrevidos.

Palabras clave— Crypt of the Pixelmancer, Inteligencia Artificial, jugador, Godot, videojuego, dificultad, roguelike, layout, Sprite, animación

Abstract— Artificial Intelligence is a central feature in the world of video games since it is responsible, through the behavior of the enemies, for generating the best possible gaming experience for the user. In this project it has been sought that the final result counts with various functions to generate a satisfactory experience when playing the game. The AI of the enemies has been implemented in such a way that their behavior patterns vary depending on the type of weapon the player is carrying. A title screen with the game's name and the map's images and an interactive option selector welcomes players upon starting a game. Along with them, a visual section has been implemented with sprites for the player with animations, sprites for enemies, as well as different level backgrounds depending on the selected map. In short, the game is ready to provide many hours of fun and a demanding challenge to the most daring players.

Index Terms—— Crypt of the Pixelmancer, Artificial Intelligence, player, Godot, videogame, difficulty, roguelike, layout, sprite, animation

1 Introducción

1.1 Motivación del proyecto

Los videojuegos son un medio de entretenimiento audiovisual y digital que, a pesar de nacer en los años 80, no ha sido hasta hace relativamente pocos años han pasado a ser parte de la vida diaria de la gente.

Durante todos estos años, han nacido una gran variedad de títulos de índole muy diferente, que podemos agrupar según sus similitudes bajo etiquetas llamadas genero de videojuegos. Estas etiquetes permiten a la gente saber cómo será un juego antes de jugarlo para así crear unas expectativas sobre él. Por ejemplo, uno de los géneros más antiguos y famosos es el género de plataformas, con personajes tan icónicos como Mario, Sonic o Spyro, cuyos juegos se caracterizan por tener que correr, saltar o escalar a través de obstáculos para llegar a la meta y poder completar el juego.

Uno de los géneros que ha ganado mucha popularidad en los últimos años es el género "souls-like", cuyo nombre referencia a la saga de videojuegos de From Software, Dark Souls. Este género es muy famoso por su dificultad y lo exigente que es con el jugador. Una de sus características más notables son las batalles con los jefes, cuya Inteligencia Artificial (I.A.) los hace especialmente difíciles. Así, lograr vencerlos genera en el jugador una sensación de triunfo y superación personal.

1.2 Objetivos del proyecto

El objetivo principal de este proyecto es conseguir un videojuego completo. Para ello, se añadirán las funcionalidades necesarias para considerar que un juego está completo. Estas funcionalidades son:

1- Programar la I.A. de los enemigos, que en este caso responderán distinto según el arma que lleve el jugador.
2- Conseguir que la generación de mapa varie en función de la selección del usuario y se creen layouts con formas predefinidas.
3- Añadir música para dar ambiente de aventura.
4- Un ataque especial al personaje jugador.

También, hay otros objetivos secundarios, como, por ejemplo, hacer que la música sea adaptativa (que se modifique cuando el jugador tenga poca vida), dotar de una estética común al juego (píxel art en este caso) y añadir diferentes niveles de dificultad.

- *E-mail de contacte: raul.salinas.natal@gmail.com*
- *Mención realitzada: Computació*
- *Treball tutoritzat per: Jorge Bernal del Nozal (Ciencias de la Computación)*
- *Curs 2022/23*

1.3 Riesgos del proyecto

Riesgo	Definición	Impacto	Contingencia
Generación de mapas repetitivos	Al definir la forma en que se genera el mapa previamente puede afectar negativamente a la experiencia del jugador, ya que una vez conozca la forma del mapa y juegue unas cuantas veces puede aburrirse de jugar siempre lo mismo	Medio	Crear el algoritmo de creación procedural de manera que no cree siempre la misma forma, preparar distintos mapas para que el usuario tenga más opciones a escoger.
Pérdida del proyecto debido a un error informático	Perdida de los documentos del proyecto debido a un fallo informático o a que se estropee el ordenador.	Alto	Mantener el proyecto en la nube y en diferentes dispositivos para poder trabajar en cualquier PC.
Niveles de dificultad mal calibrados	La versión fácil puede ser demasiado aburrida y la difícil demasiado frustrante	Medio	Realizar diversas pruebas para encontrar un buen equilibrio
Actualización importante de la herramienta Godot	Godot podría actualizarse a una nueva versión y generase problemas con el proyecto	Bajo	Trabajar en una misma versión de Godot todo el proyecto.
Ser demasiado ambicioso con las nuevas funcionalidades	Al intentar añadir muchas funcionalidades nuevas, no ser capaz de acabar el trabajo a tiempo.	Crítico	Realizar múltiples reuniones con el tutor para mantener un buen seguimiento y reconducir si fuera necesario

1.1. Requisitos del proyecto

Para crear un videojuego es necesario utilizar un motor que nos provea de un entorno con una serie de rutinas de programación predefinidas. Estas rutinas son las que permiten diseñar el videojuego y definen el sistema de físicas que se utilizará. De entre los motores de videojuegos actuales, hay algunos que son más famosos debido a la cantidad de funciones y facilidades que aportan. Dos de los más destacados son Unity [1] y Unreal Engine [2], siendo usados hasta en los videojuegos triple A de las mejores desarrolladoras de videojuegos actuales.

En este proyecto se ha decidido usar Godot [3], un motor de videojuegos libre y de código abierto que permite diseñar videojuegos en 2D y en 3D y es funcional en casi todos los sistemas operativos actuales. Pese a no ser el motor de videojuegos con más renombre o el más utilizado, debido a su funcionamiento con nodos y las facilidades que aporta al desarrollo en 2D se ha considerado la mejor opción para este proyecto.

Los requisitos necesarios para ejecutar el videojuego resultado del proyecto se pueden separar en requisitos hardware y requisitos software, listados a continuación:

Hardware

El hardware mínimo necesario para poder correr Godot y, por tanto, el juego es:

- Soporte para OpenGL 2.1
- 4 GB de RAM.
- Cualquier CPU de, al menos, dos núcleos
- 1 GB de almacenamiento

Software

Los requisitos del software son:

- Windows 7 o más reciente, macOS 10.10 o más reciente, Linux (64-bit o 32-bit x86)
- Godot 3.4.4
- OpenGL 3.3

1.2. Diagrama de Gantt

El desarrollo del proyecto se divide en distintas fases, explicadas a continuación:

- Planificación: En esta fase inicial del proyecto se han de definir los objetivos que se querían asumir antes de la entrega y se ha organizado temporalmente las tareas necesarias para completarlos.

- Investigación: Durante esta fase, se ha hecho el proceso de aprendizaje necesario para llevar a cabo el desarrollo del proyecto. Las principales fuentes de estudio han sido la documentación online de Godot [4], video tutoriales de Youtube [5], [6], [7] y "*Godot Engine Game Development Projects*".

- Desarrollo de funcionalidades: Esta fase incluye todo el proceso de programación llevado a cabo durante las semanas de desarrollo del proyecto, siendo la fase más duradera de todas.

- Apartado visual: En esta fase se ha trabajado en el apartado gráfico del juego, creándose las distintas imágenes y animaciones.

- Test y encuestas de usuarios: Esta fase del proyecto es de las últimas en llevarse a cabo ya que es necesario tener el juego en un estado final. Se comparte el juego a distintos usuarios y se les pide que lo

valoren. Con los resultados obtenidos podemos comprobar el estado del juego a ojos ajenos y sacar conclusiones o apartados a mejorar de cara al futuro.

La distribución temporal de las distintas fases se puede apreciar de manera visual en el Diagrama de Gantt en el apéndice A1.

1.3. Metodología

La metodología que se seguirá en este proyecto será Kanban [8] para tener un referente visual de las tareas realizar y así poder organizar mejor el trabajo a hacer y ver cómo va avanzando el proyecto. Junto a este método, se realizarán tutorías con el tutor para revisar el avance del proyecto y corregir el trabajo hecho, el ritmo al que se avanza o el planteamiento de las tareas a realizar.

1.4. Estado del arte

Crypt of the Pixelmancer es un juego con claras referencias a otros títulos y comparte las características típicas de los juegos *rogue-like*, algunos de los mayores títulos de este género son: *The Binding of Isaac* [Fig. 1], *Dead Cells* o *Enter the Gungeon*. El nombre del género proviene del videojuego *Rogue*, desarrollado en 1980, siendo éste el primero en proponer las mecánicas que caracterizan este género.

Fig.1 The Binding of Isaac

Todos estos juegos destacan por tener una creación de mundos procedural, jefes de nivel y diferentes tipos de habilidades para lidiar con nuestros enemigos. Es de estos títulos de los que se ha tomado inspiración para añadir las nuevas funciones al juego que mejorarán la experiencia de juego. Además, todos ellos han sido ampliamente alabados tanto por la crítica, como por la comunidad de jugadores.

Por ejemplo, en el juego the *Binding of Isaac*, el protagonista debe vagar por todo el nivel en busca de la sala del jefe. Para avanzar por las diferentes salas primero se debe eliminar a todos los enemigos que se encuentran en ella mediante ataques que pueden ser disparados de manera ortogonal. Una vez que el personaje muere debe volver a empezar de 0 y volver a explorar el mapa, que en esta nueva partida es totalmente distinto a la anterior.

En *Dead Cells* por otro lado, existen multitud de armas que puedes ir cambiando a medida que avanzas en los niveles, pudiendo empezar con una espada, un arco, un martillo, entre otras. Eso hace que cada partida, además de tener el mapa distinto, el modo de juego sea diferente en función de las armas que consigas.

2 Desarrollo

2.1 Estructura general del proyecto

En este apartado se explicará cómo está estructurado el código y cómo se relacionan los diversos ficheros entre sí.

Como se ha mencionado en apartados anteriores, este proyecto ha sido desarrollado usando Godot como motor. Éste, permite diversos lenguajes de programación, pero como lenguaje principal utiliza GDScript, que es el que mejor aprovecha las capacidades del entorno y es con el que se ha programado el juego.

Godot funciona mediante nodos y escenas. Tanto las escenas como los nodos tienen dos partes, la representación visual y la parte de código. En la parte visual es donde se trabajan los aspectos visuales del nodo o de la escena y en la parte de código se programan los comportamientos de los distintos objetos.

Un nodo es una pieza concreta que se crea con un propósito concreto. Por ejemplo, el personaje jugable es un nodo donde en su parte visual están los sprites y animaciones y el código dicta qué puede o no hacer.

Una escena se rellena con nodos para crear un escenario. Por ejemplo, la pantalla de selección de opciones cuenta con un nodo para el fondo, con botones que cambiarán el tipo de partida que se jugará y con un botón de inicio que lleva a otra escena donde se ve la sala inicial, con un fondo predeterminado diferente [Fig. 2] si el jugador utiliza teclado o [Fig.3] si el jugador juega con mando, y el nodo del personaje.

Fig.2 Fondo con teclado

Fig.3 Fondo con mando

2.2 Menú de inicio y selección de opciones

Al iniciar el juego, aparece una pantalla de título [Fig. 4] con un botón que lleva a la selección de las opciones de creación de partida. En esta pantalla se puede apreciar el título del juego *Crypt of the Pixelmancer* y un fondo de pantalla que representa los 4 mapas seleccionables.

Fig. 4 Pantalla de inicio de juego

Fig. 5 Pantalla de selección de opciones

Como se puede apreciar en la figura [Fig. 5] , la pantalla de selección de opciones permite elegir dificultad (entre "Easy", "Normal" y "Hard") y un tipo de mapa (a escoger entre "Desert", "Tower", "Volcano" y Mountain). La selección de mapa se refleja oscureciendo la imagen del mapa seleccionado. Por defecto el mapa seleccionado es "Desert" y la dificultad es "Normal", tal y como aparece en la imagen.

La elección de la dificultad modifica algunos parámetros de los enemigos, como la velocidad de movimiento, la velocidad de ataque, la cadencia de disparo o el daño que hace al jugador, para que resulte más fácil o difícil la partida en función de lo escogido. Esta elección también influye en la cantidad de enemigos que aparecen por sala y en la velocidad con la que se carga el ataque final.

2.3 Mapas

El selector de mapa permite escoger la ambientación donde se llevará a cabo la partida afectando al fondo y a cómo se genera el mapa.

La primera opción, "Desert", representa las arenas de un desierto [Fig. 6 a] [Fig. 6 b] y tiene una generación de mapa procedural lo cual hace que la disposición de las salas cambie en cada partida. Esto representa lo confuso que puede ser el desierto y provee al juego de un mapa que varía cada partida, aportando rejugabilidad al juego

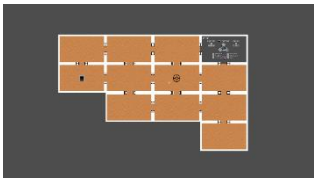


Fig.6 a Layout Desert ejemplo 1

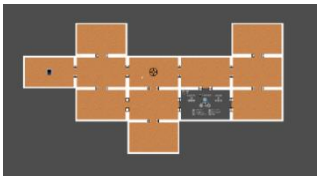


Fig. 6 b Layout Desert ejemplo 2

La segunda opción, "Tower", está basada en una torre, con el fondo representando un suelo enladrillado y el mapa se construye hacia arriba exclusivamente [Fig. 7].

Fig.7 Layout Tower

La tercera opción, "Volcano", tiene un fondo de tonos rojizos y marrones representando el suelo de un volcán y se construye de manera descendiente, representando el descenso al centro del volcán [Fig. 8].

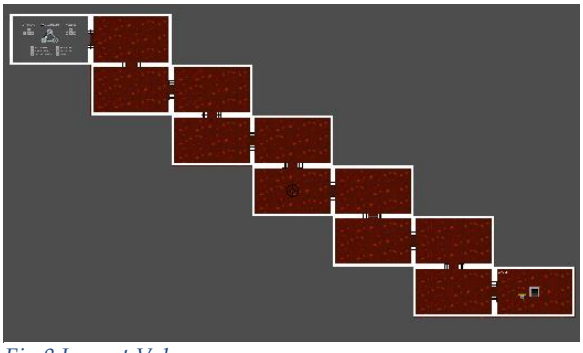


Fig.8 Layout Volcano

La última opción, "Mountain", representa la escalada a la cima de una montaña nevada, con tonos azules y blanquecinos y se construye de manera ascendente [Fig.9].

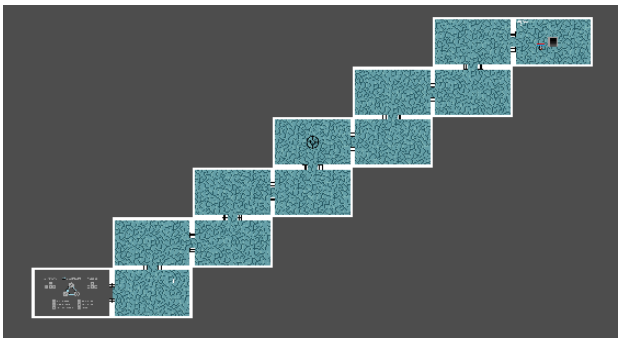


Fig.9 Layout Mountain

2.4 Jugador

En este apartado se hablará del nodo del personaje jugador, diferenciando entre su parte de código y su parte visual.

2.4.1 Características

La principal característica del nodo del jugador es que cuenta con 3 tipos distintos de ataques: uno cuerpo a cuerpo (representado con un guerrero [*Fig. 14 a*]), uno de distancia media y alta velocidad (representado con un mago [*Fig. 14 b*]) y uno de distancia larga y velocidad media (representado con un arquero [*Fig. 14 c*]).

Este factor es el núcleo de la jugabilidad y la clave para pasarse el juego ya que permite adaptar nuestra estrategia para hacer frente a los distintos enemigos o al estilo que sea más cómodo al usuario. El usuario puede mover al personaje y atacar de manera ortogonal, aunque se pueden conseguir desplazamientos diagonales presionando dos teclas o mediante un mando.

Fig. 14 Sprites del personaje

Como se puede apreciar en las imágenes anteriores, encima del Sprite hay tres barras, una roja que indica la vida restante del jugador, una barra azul que indica la resistencia (cada ataque que haga el usuario reduce un poco esa barra, para recuperarla, debe dejar de atacar) y una última barra que indica el estado de carga del ataque final (la barra empieza vacía y se va rellenando de color verde a medida que se derrotan enemigos).

En función del enemigo derrotado se ganan más o menos puntos de carga. Por ejemplo, derrotar a un Bouncer incrementa la barra en un 10%, un Spin en cambio la rellena un 30%. Una vez la barra está totalmente llena se puede apretar la barra espaciadora del teclado o el botón R3 del mando para vaciar de nuevo la barra y ejecutar un poderoso ataque final [*Fig. 15*].

Una vez pulsado el botón caen del cielo bolas de fuego mágico que eliminan a los enemigos de la sala. En el caso del enemigo final, éste perderá 3 de sus 10 vidas.

Fig. 15 Ataque final del jugador

2.4.2 Aspectos visuales

Para el aspecto visual del personaje se ha querido enfatizar el hecho de que cada arma es distinta, es por eso que cada uno de los tres diseños implementados tiene un código de colores distinto, para que sea más sencillo para el usuario diferenciarlas y que sepa que arma esta usando en cada momento.

Además de los diseños que se han mostrado en el apartado anterior, se han añadido tres animaciones al personaje para cada una de las tres armas [*Tabla 9*], una idle para cuando el jugador está quieto, una para cuando está andando y una última para cuando está atacando. Estas animaciones son posibles gracias a que Godot tiene una opción que crea sprites animados a partir de una secuencia de imágenes y la velocidad en frames por segundo a la que se reproducirá la animación. Para evitar tener el doble de diseños, uno para cada dirección, cuando cambia la dirección al personaje, se invierte el eje horizontal para que el personaje mire hacia la dirección correcta.

Guerrero	Idle	
	Andar	
	Ataque	
Mago	Idle	
	Andar	
	Ataque	
Arquero	Idle	
	Andar	
	Ataque	

Tabla 9 Frames de animaciones del jugador

2.5 Enemigos

En este apartado veremos la implementación de la IA de los diferentes enemigos, que cambiarán sus patrones de comportamiento en función del arma con la que vaya equipada el personaje [Tabla 1] [Tabla 3] [Tabla 5] [Tabla 7].

Los enemigos son más agresivos (tendrán una velocidad de movimiento mayor para atacar al jugador) si el jugador usa el arma cuerpo a cuerpo o más poderosos (sus ataques producen más daño a la salud del jugador) si el jugador usa armas a distancia. Los cambios concretos de cada tipo de enemigo se detallan en los siguientes apartados [Tabla 2] [Tabla 4] [Tabla 6] [Tabla 8].

2.5.1 Turret

Tal y como se puede apreciar en la [*Fig. 10*], este enemigo está representado como una ballesta que dispara una flecha cada vez que carga su barra, detectando la posición del personaje y rotando para acertarle. Además, puede defenderse de los ataques a distancia que vengan de frente, haciendo que solo pueda ser derrotado mediante un ataque cuerpo a cuerpo o de un disparo desde un lateral o desde detrás. Conseguir esto último es difícil, ya que rota para encarar al jugador cada vez que dispara.

Armas cuerpo a cuerpo	Disparará un poco más rápido y rotará más. Esto hará que el jugador tenga que esquivar más a medida que se acerca
Armas a distancia	Dispara menos y rotará con menor frecuencia. Esto facilitará al jugador rodearlo para acertarle con un proyectil.

Tabla 1 Modificaciones del patrón de Turret según arma

Fácil	- Menor daño al jugador - Menor cadencia de disparo - Rota más lento
Normal	- Mantiene unos valores equilibrados
Difícil	- Mayor daño al jugador - Mayor cadencia de disparo - Rota más rápido

Tabla 2 Modificaciones del patrón de Turret según dificultad

2.5.2 Kamikaze

Este enemigo persigue al jugador para hacerle daño mediante contacto. Cada vez que recibe un impacto aumenta su velocidad hasta recibir el tercer impacto, en el que es derrotado. Si consigue alcanzar al jugador antes de recibir los 3 golpes explotará haciendo mucho daño a la salud del jugador. Debido a su comportamiento, este enemigo está representado como una punta de flecha con el símbolo de peligro [*Fig. 11*].

Armas cuerpo a cuerpo	Se mueve más rápido al inicio, pero ganará menos velocidad con cada impacto.
Armas a distancia	La velocidad inicial será menor, pero aumentará más con cada golpe

Tabla 3 Modificaciones del patrón de Kamikaze según arma

Fácil	- La velocidad se mantiene constante
Normal	- Mantiene unos valores equilibrados
Difícil	- Mayor daño al jugador - Mayor aumento de velocidad

Tabla 4 Modificaciones del patrón de Kamikaze según dificultad

2.5.3 Bouncer

Este enemigo se mueve en un eje vertical u horizontal y dispara flechas al jugador cuando lo tiene en frente. Para reflejar su comportamiento, este enemigo está representado como una estrella de cuatro puntas [*Fig. 12*].

Armas cuerpo a cuerpo	Se mueve más rápido y dispara menos
Armas a distancia	Se mueve más despacio y dispara más seguido

Tabla 5 Modificaciones del patrón de Bouncer según arma

Fácil	- Menor daño al jugador - Menor velocidad de movimiento - Menor cadencia de disparo
Normal	- Mantiene unos valores equilibrados
Difícil	- Mayor daño al jugador - Mayor velocidad de movimiento - Mayor cadencia de disparo

Tabla 6 Modificaciones del patrón de Bouncer según dificultad

2.5.4 Spin

Este enemigo varía su comportamiento según la distancia con el jugador además de tener en cuenta el arma que éste lleve equipada. Cuenta con escudos para protegerse de los ataques a distancia y con una barra de vida que se regenera cuando está cerca de acabarse. Es el enemigo común más complejo de los que hay en el juego y es por ese motivo que aparece en las últimas salas. Está representado con un cuadrado que indica el modo de ataque y rodeado de escudos que desaparecen al recibir un impacto [*Fig. 13*].

Armas cuerpo a cuerpo	Dispara un poco más despacio, se moverá más rápido y rotará menos sus escudos.

Armas a distancia	Se mueve más despacio, disparará más rápido y rotará más rápido sus escudos para protegerse de los proyectiles.

Tabla 7 Modificaciones del patrón de Spin según arma

Fácil	- Menor daño al jugador - Menor cadencia de disparo - Rota más lento sus escudos
Normal	- Mantiene unos valores equilibrados
Difícil	- Mayor daño al jugador - Mayor cadencia de disparo - Rota más rápido sus escudos

Tabla 8 Modificaciones del patrón de Spin según dificultad

2.5.5 Sprites

Fig.10 Sprite Turret

Fig.11 Sprite Kamikaze

Fig.12 Sprite Bouncer

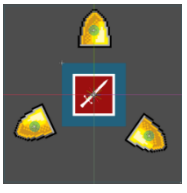


Fig.13 Sprite Spin

2.6 Batalla final

La batalla contra el boss final es una batalla que cuenta con diversas fases donde hay que utilizar las distintas armas para poder ser derrotado.

En la primera fase [*Fig. 16*] el boss avanzará en círculos por la pared de la sala mientras dispara proyectiles. En este estado es inmune a ataques a distancia por lo que hay que pelear cuerpo a cuerpo para bajarle vidas.

Una vez empieza la segunda fase [*Fig. 17*], el boss cuenta con un escudo que le protege del daño a menos que se le golpee a mucha velocidad, cosa que obliga a atacar usando el ataque mágico.

En la última fase [*Fig. 18*], el boss desaparece y crea una grieta en medio de la sala, separándose del jugador y obligándole a usar el arma a distancia, ya que es la única con rango suficiente para alcanzarle.

Fig. 16 Primera fase de la batalla final

Fig. 17 Segunda fase de la batalla final

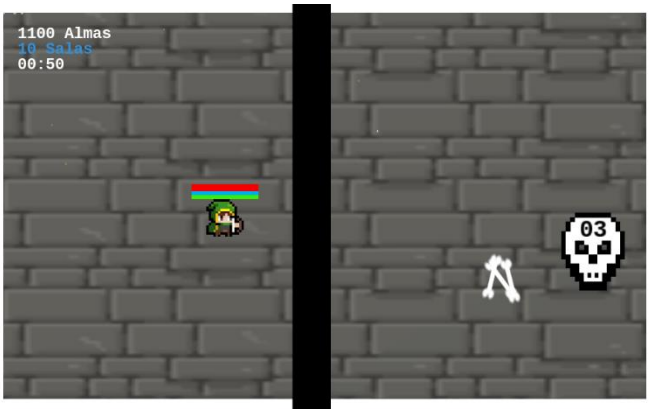


Fig. 18 Última fase de la batalla final

2.7 Música adaptativa

En el juego hay una música ambiente de fondo durante las partidas y se ha creado una función que hace sonar una música de alarma cunado la vida del personaje está por debajo del 20% de la vida máxima. Se vuelve a la música original una vez el personaje se cura en la hoguera.

2.8 Resumen de mejoras respecto al original

En este apartado se hará una comparativa de los cambios realizados en este proyecto con sus respectivos referentes en el proyecto anterior a este [*Tabla 10*].

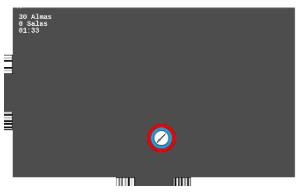
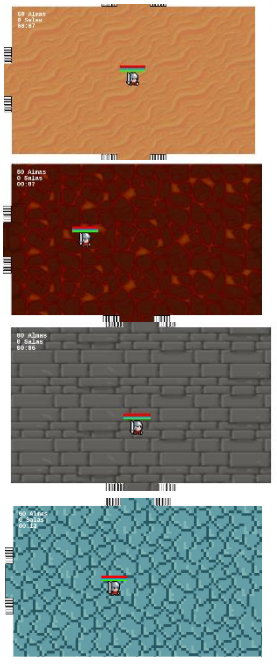
Proyecto original	Proyecto final
Carece de pantalla inicial	Tiene pantalla inicial [Fig. 4]
Carece de pantalla de opciones	Tiene pantalla de opciones [Fig. 5]
1 opción de mapa	4 opciones de mapas
No tiene selector de dificultad	Tiene 3 dificultades distintas para escoger
Enemigos con un único patrón de comportamiento	Gracias a los cambios en la I.A. de los enemigos, varian su comportamiento en función del arma equipada por el jugador. Tienen dos comportamientos principales, cuando el usuario usa el guerrero se mueven más rápido para intentar hacer daño al jugador. Usando las armas a distáncia, el mago o el arquero, los enemigos se mueven más despació pero cada impacto hace más daño a la salud del jugador.
Sin música	Con música adaptativa
Interfaz circular rodeando al jugador	Interfaz en barras encina del jugador
Interfaz con dos círculos	Interfaz con tres barras
No tiene animaciones para el jugador	Tiene 3 animaciones distintas para cada tipo de arma [Tabla 9]
No tiene ataque final	Tiene ataque final [Fig. 15]
Salas sin background 	Backgrounds distintos 

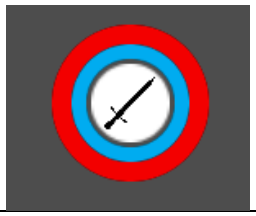
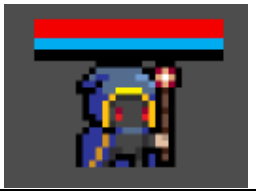
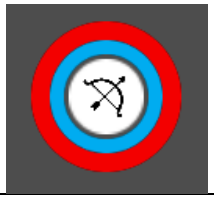
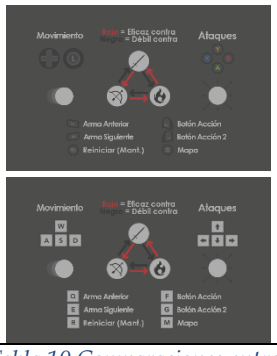
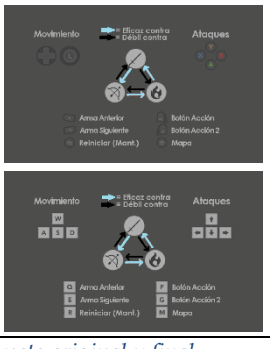
Imagen del guerrero 	Imagen del guerrero 
Imagen del mago 	Imagen del mago 
Imagen del arquero 	Imagen del arquero 
Sala inicial 	Sala inicial 

Tabla 10 Comparaciones entre proyecto original y final

3 RESULTADOS

Para comprobar si *Crypt of the Pixelmancer* se encuentra en un punto que se pueda considerar una experiencia completa, se ha recopilado la opinión de un grupo de jugadores de prueba, o "betatesters" para obtener su feedback del juego en el estado actual y valorar si las mejoras respecto al juego original aportan una sensación positiva a la experiencia de juego.

Estos jugadores han experimentado por primera vez *PoligonSouls* 2.0 (título del proyecto original) y *Crypt of the Pixelmancer* sin tener ayuda por parte del autor. Una vez han experimentado ambos juegos durante el tiempo que han creído suficiente, han contestado un formulario dando su opinión sobre las funcionalidades añadidas y sobre la experiencia vivida, además de contar con un apartado para reportar bugs, en caso de que los hayan experimentado.

En el apartado A2 del apéndice, se pueden ver las gráficas de los resultados obtenidos del formulario

Teniendo en cuenta estos resultados, podemos decir que a nivel usuario la experiencia resulta divertida y presenta una mejora reflejada en las notas medias puestas por los usuarios, pasando de tener una media de "6,5" a "8" [*Fig. 19*] [*Fig. 20*].

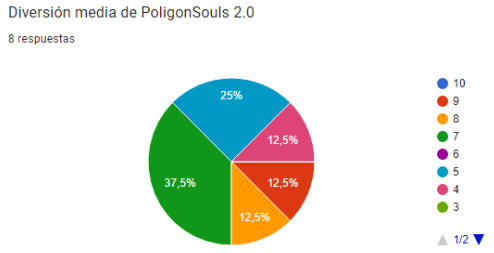


Fig. 19 Gráfica de diversión media de PoligonSouls 2.0

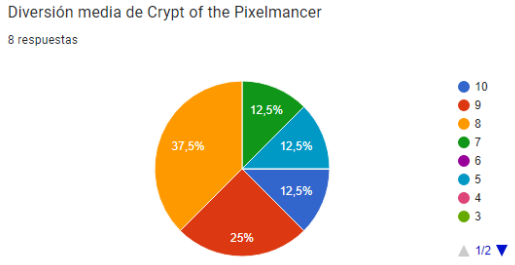


Fig. 20 Gráfica de diversión media de Crypt of the Pixelmancer

También se puede observar una mejora en la media de la claridad de juego. Esto es debido a todas las mejoras gráficas que se han implementado y permiten tener una mejor comprensión del juego, tal y como reflejan las siguientes gráficas [*Fig. 21*] [*Fig. 22*].

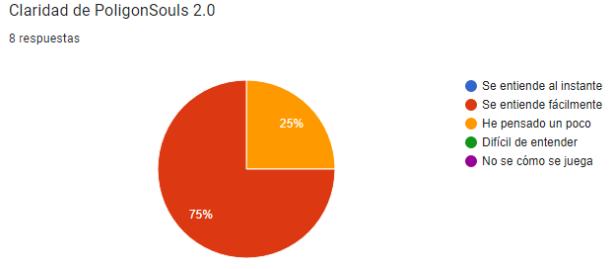


Fig. 21 Gráfica de claridad de PoligonSouls 2.0

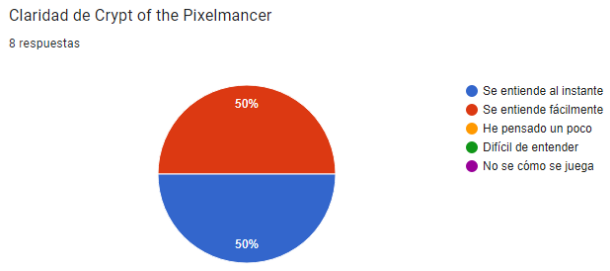


Fig. 22 Gráfica de claridad de Crypt of the Pixelmancer

La dificultad del juego es un tema muy importante en este proyecto y la adición de los modos de dificultad supone una reducción del número de personas que opina que el juego es demasiado difícil y une a la mayoría de gente a escoger la opción de que está bien diseñado o que el juego es difícil, tal y cómo se puede apreciar en las siguientes gráficas [*Fig. 23*] [*Fig. 24*].

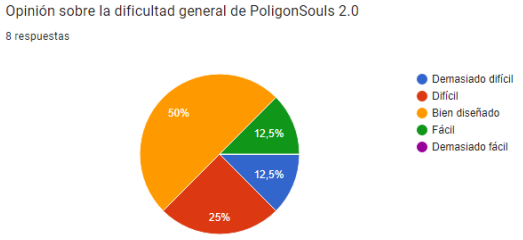


Fig. 23 Gráfica de dificultad general de PoligonSouls

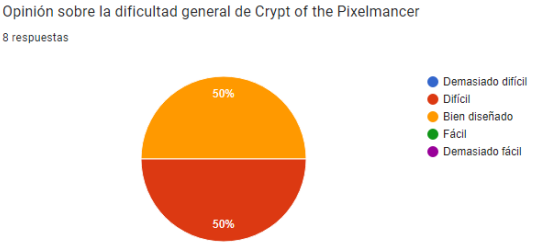


Fig. 24 Gráfica de dificultad general de Crypt of the Pixelmancer

De las respuestas recibidas en la pregunta abierta se han separado en dos categorías: opiniones generales del juego (algunas de ellas incluidas en el Apéndice A2) y propuestas de mejoras y bugs (las cuales han sido incluidas en el punto del informe *4.3 Trabajo futuro y mejoras*).

Se ha creado un video de presentación dónde se muestran las principales características del juego, así como algo de gameplay [6]. También se puede acceder al código del juego mediante el enlace de GitHub [7].

4 Conclusiones

4.1 Conclusiones generales del proyecto

Actualmente *Crypt of the Pixelmancer* se encuentra en un estado funcional y jugable, con un apartado gráfico cohesivo. Se puede considerar una versión 1.0 de un juego y en sí, está acabado, pero aún se le pueden añadir algunas funcionalidades más para sumar valor al producto final.

Aun así, se puede afirmar que el juego está en un estado muy bueno, ya que los distintos usuarios que han probado el juego han dado opiniones muy positivas sobre el estado actual del producto final, reflejadas en la nota media extraída de la gráfica [*Fig. 20*]. El resto de las gráficas sugieren una conclusión similar.

El objetivo principal del proyecto es poder proporcionar una experiencia completa y satisfactoria a los usuarios que

jueguen a *Crypt of the Pixelmancer*. Para ello se han implementado todas las funciones necesarias que hacen de un videojuego una experiencia completa. Se ha implementado Inteligencia Artificial a los enemigos, y se ha hecho que respondan con un patrón de comportamiento distinto en función del arma del jugador. Para evitar que el jugador se aburra de jugar siempre al mismo juego, se han implementado diferentes mapas, tres con un Layout predefinido y uno con generación procedural para aportar rejugabilidad y variedad. Por último, se ha trabajado en el apartado estético, manteniendo una temática común dentro del juego mediante los sprites y las animaciones. Con todo esto, se puede concluir que el proyecto ha cumplido su propósito de manera satisfactoria.

4.2 Aprendizaje personal

Este proyecto ha supuesto para mí un reto que estaba deseoso de afrontar, el mundo de los videojuegos ha sido una parte troncal de mi vida desde que era pequeño y lo cierto es que hace tiempo que tengo ganas de poder adentrarme en ese mudo de manera profesional. En ese sentido, creo que desarrollar este proyecto me ha permitido echar un pequeño vistazo a ese mundo y me ha proporcionado, aunque de manera parcial, algunas herramientas que puede que me permitan acceder a él en un futuro.

Durante el desarrollo de este proyecto he tocado múltiples ámbitos del desarrollo de videojuegos que jamás había tocado antes, el mundo de las animaciones y el píxel art es algo que no había tocado nunca, pero puedo afirmar que es una de las mejores experiencias que me llevo del desarrollo y la que más he disfrutado de hacer.

4.3 Trabajo futuro y mejoras

Crypt of the Pixelmancer ya proporciona una experiencia completa y cierta rejugabilidad con sus diversos mapas y la generación procedural del mapa de desierto. No obstante, aún se pueden añadir o modificar funciones para crear una experiencia aún mejor.

Una de las principales propuestas que se ha tenido presente durante el desarrollo del proyecto ha sido la creación de un método que permita mantener una sesión, manteniendo un registro de las partidas que ha jugado un mismo usuario. Eso permitiría mantener un control sobre los niveles que se ha pasado una persona y, al pasarse los cuatro mapas, desbloquear un quinto mapa. Otra opción que se ha barajado de un estilo similar es encadenar los mapas, en este supuesto una vez se llega a la sala final del mapa seleccionado no se pelearía contra el boss final, sino que te llevaría de regreso a la pantalla de selección de mapas para escoger un segundo mapa (bloqueando el mapa ya pasado), y seguir completando los mapas hasta completar los 4 mapas antes de pelear contra el boss final.

De entre los comentarios recibidos por los usuarios se han encontrado algunas propuestas de mejora. Entre ellas se ha propuesto adecuar mejor la animación de ataque de los personajes, ya que actualmente y debido a la velocidad de ataque se puede ver una vez cada dos disparos. Otra mejora sugerida ha sido la de añadir efectos de sonido al recibir un impacto enemigo, así como una pequeña animación para dejar más claro cuándo se pierde vida.

AGRADECIMIENTOS

En primer lugar, me gustaría dar las gracias tanto a Xiang Lin, compañero y predecesor en el desarrollo de este proyecto, como a Jorge Bernal, mi tutor en este Trabajo de Fin de Grado, sin su ayuda y consejos no habría podido trabajar en este proyecto que he disfrutado tanto ni el estado final sería el que es hoy día.

Por otro lado, me gustaría dar las gracias a todas aquellas personas que me han ayudado y aguantado durante el desarrollo del mismo, en especial quiero hacer mención a Maria Tomàs, que sin su inestimable apoyo y ayuda durante esta dura época no me habría sido posible hacer este trabajo. También a Pol Medina, que me ayudó en las primeras fases del proyecto a definir el apartado visual y aportó varias ideas que han llegado hasta la versión final del juego. Muchas gracias por todo.

Por último, me gustaría agradecer a todas aquellas personas que probaron el juego a modo de betatesters y me han ayudado al reportar sus experiencias y dando sus opiniones en el formulario.

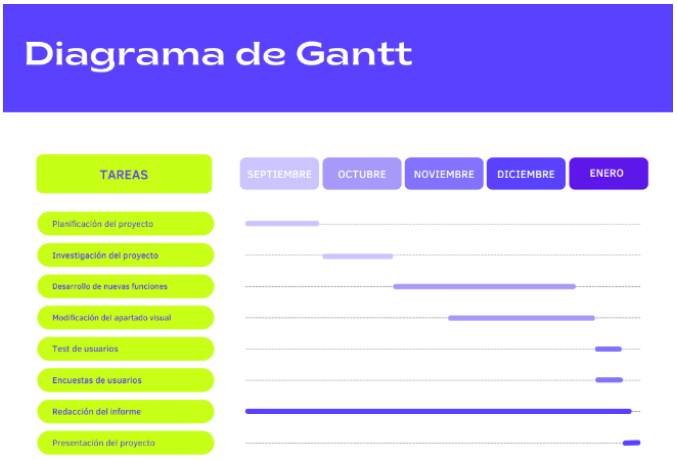
BIBLIOGRAFÍA

- [1] Unity Engine (2023, Feb. 1). Página web de Unity [Online]. Disponible: https://unity.com
- [2] Unreal Engine (2023, Feb. 1). Página web de Unreal Engine [Online]. Disponible: https://www.unrealengine.com/es-ES
- [3] Godot Engine (2023, Feb. 1). Página web de Godot Engine [Online]. Disponible: https://godotengine.org
- [4] Godot Docs. (2022, Mar. 1). Documentación de Godot [Online]. Disponible: https://docs.godotengine.org/es/stable/index.html
- [5] A. Céspedes. (2018, Oct. 22). Como crear un juego simple 2D en Godot [Online]. Disponible: https://www.youtube.com/watch?v=vFbkF0JhSuI&list=PL5K_XeigIfdJeaJ3-_YgnzD711StmJkZh&index=1
- [6] GrandMasterTaco. (2021, Ene. 9). DIY Procedural Generation - Quick Python [Online]. Disponible: https://www.youtube.com/watch?v=NUXHw3DBoq8
- [7] C. Bradfield. Godot Engine Game Development Projects. Los Ángeles: Packt Publishing, 2018
- [8] Kanbanize (2022, Jun. 26) Qué es Kanban: Definición, Características y Ventajas. [Online] Disponible: https://kanbanize.com/es/recursos-dekanban/primeros-pasos/que-es-kanban
- [9] Raül Salinas Natal (2023, Feb. [X]) Crypt of the Pixelmancer Gameplay [Online] Disponible: https://youtu.be/di8wHH9tHdE
- [10] Raül Salinas Natal (2023, Feb. 6) GitHub de Crypt of the Pixelmancer [Online] Disponible: https://github.com/Raul-SalinasNatal/Crypt_of_the_Pixelmancer.git

APÉNDICE

A1. DIAGRAMA DE GANTT

Aquí se puede ver el diagrama de Gantt del apartado 1.5 de este informe.



A2. SECCIÓ D'APÈNDIX

Gráficas del resultado del formulario

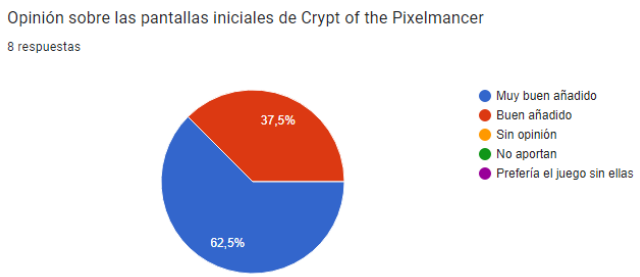


Fig. 25 Opinión sobre las pantallas iniciales

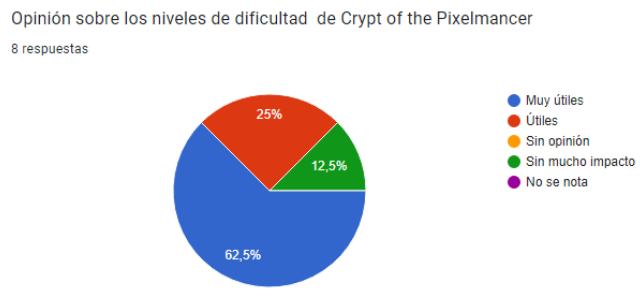


Fig. 26 Opinión sobre los niveles de dificultad

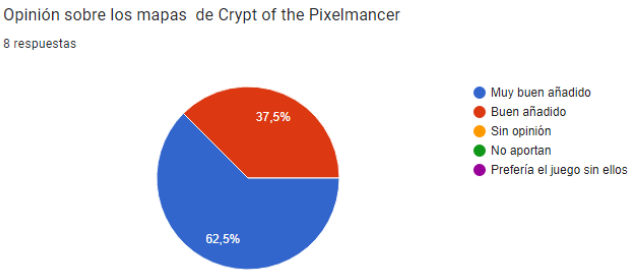


Fig. 27 Opinión sobre los mapas nuevos

Fig. 28 Opinión sobre la actualización visual del personaje jugable

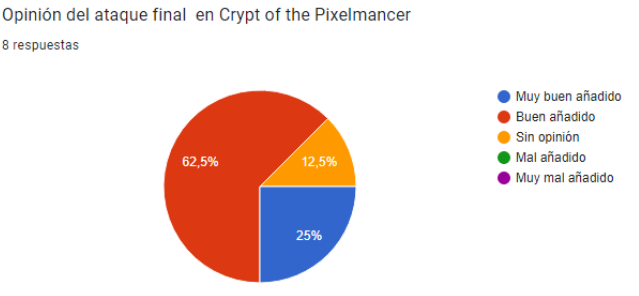


Fig. 29 Opinión sobre el ataque final añadido al jugador

A3. SECCIÓ D'APÈNDIX

Comentarios de usuarios.

- "Mucho más atractivo visualmente y más claro. Los niveles evitan la frustración del jugador, pudiendo jugar a un nivel dificil del usuario pero posible."

- "Me ha gustado el añadido de la musiquita, le da un toque"

- "A pesar que es un estilo de juego no es de mi tipo, su funcionalidad y su sencilleza hizo que jugara bastante tiempo. Sobretodo al Crypt of the Pixelmancer y su diseño de personajes y entornos. Lo hizo muy atractivo e invitaba a seguir jugando porque resultaba mas entrentenido que el PoligonSouls2.0!"